STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

Tomáš Hudeček

Mladá Boleslav 2022

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA  
MLADÁ BOLESLAV

**ROČNÍKOVÁ PRÁCE**

**Autor: Tomáš Hudeček  
Studijní obor: 18-20-M/01 Informační technologie  
Vedoucí práce: Jan Till**

Mladá Boleslav 2022

# Obsah

[Obsah 3](#_Toc101985897)

[1 Úvod 5](#_Toc101985898)

[2 Discord bot 6](#_Toc101985899)

[2.1 Vytvoření a spuštění 6](#_Toc101985900)

[2.2 Komunikace se serverem 6](#_Toc101985901)

[2.3 Příkazy 7](#_Toc101985902)

[2.3.1 Čtení příkazů 7](#_Toc101985903)

[2.3.2 Funkcionalita příkazu 8](#_Toc101985904)

[3 Závěr 11](#_Toc101985905)

[4 Přílohy 12](#_Toc101985906)

[4.1 Seznam obrázků 12](#_Toc101985907)

[4.2 Zdroje 12](#_Toc101985908)

**Prohlášení**

Prohlašuji, že jsem svou ročníkovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v přiloženém seznamu.

Nemám závažný důvod proti zpřístupňování této ročníkové práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Mladé Boleslavi dne podpis:

# Úvod

Smyslem této práce bylo vytvořit systém který je schopen propojit Discord[[1]](#footnote-1), což je komunikační platforma pro komunity, se známou hrou s otevřeným světem Minecraft[[2]](#footnote-2). Hlavním účelem bylo dát možnost uživateli na Discordu interagovat se serverem, který běží na Minecraftu.

Takový server si lze zakoupit online, nebo nahostovat[[3]](#footnote-3) na domácím počítači nebo notebooku. Do Minecraft serveru lze přidat plugin[[4]](#footnote-4), který využívají programátoři pro změnu designu a mechaniky her, dává jim veškerou moc manipulovat se serverem na kterém je plugin vložen.

Discord je, jak už je zmíněno, komunikační platforma, na kterém si může uživatel vytvořit server. Na tento server se mohou napojit další uživatelé. V takovém serveru se nacházejí kanály, kde si mohou členové psát nebo spolu volat. Na server si může vlastník napojit takzvaného Discord bota, což je aplikace řízená umělou inteligencí, která je schopná všeho na co je naprogramovaná.

Vytvořením těchto dvou aplikací, které spolu navzájem komunikují je zajištěno správné fungování systému.

Principem systému je, že bot vytvoří příkazy, které pak může člen na Discordu serveru, kde je bot vložen, napsat. Tím pošle žádost na propojený plugin na Minecraft serveru pro zpuštění příslušného události.

# Discord bot

Zvolený softvérový systém pro vytvoření a spuštění bota byl zvolen Node.js a psaný byl v jazyce javascript. Bot byl pojmenován McBot.

## Vytvoření a spuštění

Pro vytvoření bota je třeba vytvořit novou aplikaci na oficiálním developer portálu Discordu. Každý bot má svůj unikátní token, díky kterému lze bot dále naprogramovat.

require('dotenv').config();

const { Client } = require('discord.js');

const client = new Client({});

client.login(process.env.TOKEN);

bot je v tomto zdrojovém kódu client na kterého se pak volá .login který zajistí spuštění bota.

process.env.TOKEN je token bota z developer portálu je ale uložen v .env souboru jelikož je nutné mít token bezpečně někde uložen. Env soubor se čte pomocí dotenv balíčku.

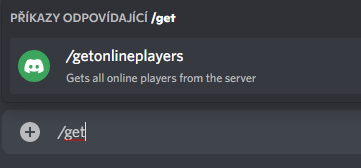
Bot se spouští v základu spouští pomocí příkazu `node index.js`. V této aplikaci je ale využit baliček nodemon, který umožňuje automatické spuštění aplikace po změně kódu tedy `npm start`.

## Komunikace se serverem

Komunikace mezi klientem a serverem probíhá na http protokolu. Discord bot se chová jako klient a v Minecraft pluginu jako http server. Komunikaci s http serverem byl v botovy použit Axios balíček. Balíček byl využit pro jednoduché vytváření http requestů.

## Příkazy

Uživatel napíše do kolonky pro zprávu název daného příkazu



Po odeslání příkazu pošle bot http požadavek na koncový bod souvisící s daným příkazem. Server buď odpoví se zprávou, že se funkce pod daným koncovým bodem provedla správně či nikoliv.



PŘÍDAT FOTO

### Čtení příkazů

Po zadání příkazu bot zjistí, zda příkaz někdo napsal pomocí metody on interactionCreate[[5]](#footnote-5) a dál zda příkaz nezadal jiný bot, to by totiž mohlo způsobit cykli a další chyby.

if (!interaction.author.bot) return;

Potom jestli příkaz existuje a na konec ho vykoná.

const command = client.commands.get(interaction.commandName);

if (!command) return;

//…

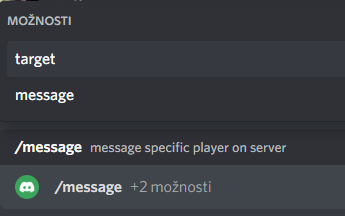
command.execute(interaction);

### Funkcionalita příkazu

Příkazy jsou uložené ve složce commands. A jednotliví příkaz je napsán v samostatném souboru.



Každý byl vytvořen pomocí exportování modulu SlashCommnadBuilder[[6]](#footnote-6). Exportuje se do spouštěcího souboru index.js, ve kterém se na základě toho zjišťuje zda napsaný příkaz existuje. V tomto SlashCommandBuilderu lze nastavit jméno a popis a to zda-li má více pod-příkazů, to znamená více argumentů v příkazu.



module.exports = {

    data: new SlashCommandBuilder()

        .setName('message')

        .setDescription('message specific player on server')

        .addStringOption(option => option.setName('target').setDescription('target to message'))

        .addStringOption(option => option.setName('message').setDescription('message to send')),

Discord názvy příkazů s mezerami s diakritikou a velkými písmeny nepřijímá.

Po zadání příkazu se spustí funkce k němu přiřazená. V tomto případě je to pokaždé GET[[7]](#footnote-7) požadavek na server pomocí Axiosu. Spolu s GET požadavkem lze poslat i params[[8]](#footnote-8) ve kterém bot posílá údaje z už zmíněných pod-příkazů.

  axios.get(`${process.env.API}/message`, {

            params: {

                target: target,

                message: message,

                author: author

            }

         }

URL adresa se nachází v .env souboru pro jednoduché přepsání při změně adresy serveru.

Po provedení se vrátí zpráva od serveru zpráva nebo chybná hláška VIZ OBRAZKY NAHORE

#### Get online players

#### Give all players cookie

# Závěr

Shrnutí práce, rekapitulace probrané problematiky. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec quis arcu leo. Quisque pretium sagittis tortor, et venenatis nisi malesuada et. Nam at consectetur enim. Proin facilisis diam placerat, pretium leo ac, imperdiet mauris.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec quis arcu leo. Quisque pretium sagittis tortor, et venenatis nisi malesuada et. Nam at consectetur enim. Proin facilisis diam placerat, pretium leo ac, imperdiet mauris.

# Přílohy

## Seznam obrázků

[Obrázek 1: Logo Qt [http://blog.updatenode.com/wp-content/uploads/2014/05/Qt\_master\_logo\_CMYK\_300dpi.png] 7](#_Toc464121377)

## Zdroje

*Dny otevřených dveří - Střední průmyslová škola Mladá Boleslav* [online]. [cit. 2016-10-17]. Dostupné z: https://www.spsmb.cz/aktuality/dny-otevrenych-dveri/

Hlad. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2016 [cit. 2016-10-17]. Dostupné z: <https://cs.wikipedia.org/wiki/Hlad>

1. Discord - <https://discord.com/> [↑](#footnote-ref-1)
2. Minecraft - <https://www.minecraft.net/> [↑](#footnote-ref-2)
3. Hostovat – server je vytvořen a poběží na vašem pc [↑](#footnote-ref-3)
4. plugin – aplikace třetí strany [↑](#footnote-ref-4)
5. intercationCreate – bot se ptá zda byl příkaz vytvořen [↑](#footnote-ref-5)
6. SlashCommandBuilder – Třída pro vytváření příkazů – součást Discord.js balíčku [↑](#footnote-ref-6)
7. GET – požadavek o získání něčeho ze serveru [↑](#footnote-ref-7)
8. Params – doplňující informace pro GET požadavek [↑](#footnote-ref-8)